

UNIVERSIDAD DISTRITAL  
FRANCISCO JOSÉ DE CALDAS

## Networks and their traffic in multiplayer games

### Redes y su Tráfico en Juegos Multijugador

### Redes e seu tráfego nos jogos multijogadores

Cristian Andrés Melo López<sup>1</sup>Octavio José Salcedo Parra<sup>2</sup>Andrés Gallego Torres<sup>3</sup>**Fecha de recepción:** diciembre 2015**Fecha de aceptación:** abril 2016

**Para citar este artículo:** Melo, C. Parra, O. & Gallego, A. (2016). Networks and their traffic in multiplayer games. *Revista Científica*, 24, 100-109. **Doi:** [10.14483/udistrital.jour.RC.2016.24.a10](https://doi.org/10.14483/udistrital.jour.RC.2016.24.a10)

#### Abstract

Computer games called multiplayer real-time, or (MCG) are at the forefront of the use of the possibilities of the network. Research on this subject have been made for military simulations, virtual reality systems, computer support teamwork, the solutions diverge on the problems posed by MCG. With this in mind, this document provides an overview of the four issues affecting networking at the MCG. First, network resources (bandwidth, latency and computing capacity), together with the technical limits within which the MCG must operate. Second, the distribution concepts include communication architectures (peer-to-peer, client / server, server / network), and data and control architectures (centralized, distributed and reproduced). Thirdly, scalability allows the MCG to adapt to changes in parameterization resources. Finally, security is intended to fend off the traps and vandalism, which are common in on-line games; to check traffic, particularly these games we decided to take the massively multiplayer game

League of Legends, a scene corresponding to a situation of real life in a network of ADSL access network is deployed has been simulated by using NS2.

Three variants of TCP, it means SACK TCP, New Reno TCP, and TCP Vegas, have been considered for the cross traffic. The results show that TCP Vegas is able to maintain a constant speed while racing against the game traffic, since it avoids the packet loss and the delays in the tail caused by high peaks, without increasing the size of the sender window. SACK TCP and TCP New Reno, on the other hand, tend to increase continuously the sender window size, which could allow a greater loss of packages and also to cause unwanted delays for the game traffic.

**Keywords:** Multiplayer, MCG, virtual reality.

#### Resumen

Los juegos de computador en tiempo real, o (MCG) están a la vanguardia de la utilización de las posibilidades de la red. Las investigaciones sobre esta materia se han hecho para las simulaciones militares,

1. Universidad Nacional de Colombia. Bogotá, Colombia. Contacto: [camelol@unal.edu.co](mailto:camelol@unal.edu.co)

2. Universidad Distrital Francisco José de Caldas. Bogotá. Colombia Contacto: [osalcedo@udistrital.edu.co](mailto:osalcedo@udistrital.edu.co)

3. Fundación Universitaria Los Libertadores. Bogotá, Colombia. Contacto: [rgallegot@libertadores.edu.co](mailto:rgallegot@libertadores.edu.co)

sistemas de realidad virtual, soporte informático del trabajo en equipo, las soluciones propuestas divergen sobre los problemas que plantea el MCG. Con esto en mente, este documento ofrece una visión general de los cuatro aspectos que afectan a la creación de redes en el MCG. En primer lugar, los recursos de redes (ancho de banda, latencia y capacidad de cálculo), junto con los límites técnicos dentro de la cual el MCG debe operar. En segundo lugar, los conceptos de distribución incluyen arquitecturas de comunicación (peer-to-peer, cliente / servidor, servidor / red), y los datos y arquitecturas de control (centralizado, distribuido y reproducido). Thirdly, la escalabilidad permite que el MCG para adaptarse a los cambios en los recursos de parametrización. Por último, la seguridad está destinado para defenderse de las trampas y el vandalismo, que son comunes en los juegos en línea; para comprobar el tráfico, en particular de estos juegos que decidimos tomar el juego masivo multijugador Liga de Leyendas, se despliega una escena de red correspondiente a una situación de la vida real en una red de acceso ADSL se ha simulado mediante el uso de NS2. Tres variantes de TCP, que significa TCP SACK, Nueva Reno TCP y TCP Vegas, han sido consideradas para el tráfico transversal. Los resultados muestran que TCP Vegas es capaz de mantener una velocidad constante, mientras que una carrera contra el tráfico juego, ya que evita la pérdida de paquetes y los retrasos en la cola causado por los altos picos, sin aumentar el tamaño de la ventana de remitente. SACK TCP y TCP New Reno, por otro lado, tienden a aumentar continuamente el tamaño de la ventana remitente, lo que podría permitir una mayor pérdida de paquetes y también para causar retrasos no deseados para el tráfico de juego.

**Palabras Clave:** Multijugador, MCG, realidad virtual.

### Resumo

O multiplayer Jogos de computador em tempo real ou (MCG) estão na vanguarda da utilização das possibilidades da rede. As investigações sobre este assunto têm sido feitas para simulações militares, sistemas de realidade virtual, suporte informatizado de trabalho em equipe, as soluções propostas divergem sobre os problemas colocados pela MCG. Com isto em mente, este

artigo fornece uma visão geral dos quatro aspectos que afetam a criação de redes no MCG. Em primeiro lugar, os recursos de redes (largura de banda, latência e capacidade de cálculo), juntamente com os limites técnicos dentro do qual o MCG deve operar. Em segundo lugar, os conceitos de distribuição incluem arquiteturas de comunicação (peer-to-peer, cliente / servidor, servidor / rede) e dados e arquiteturas de controle (centralizado, distribuído e replicado). Thirdly, a escalabilidade permite que o MCG para se adaptar às mudanças em recursos de parametrização. Finalmente, a segurança destina-se a defendê-la contra as armadilhas e vandalismo, que são comuns em jogos on-line; para verificar o tráfego em particular destes jogos, decidimos tomar o enorme jogo multiplayer League of Legends, ele é implantado uma cena rede correspondendo a uma situação da vida real em uma rede de acesso ADSL foi simulada usando NS2. Três variantes do TCP, isso significa Sack TCP, New Reno TCP e TCP Vegas, foram considerados para o tráfego de cruz. Os resultados mostram que o TCP Vegas é capaz de manter uma velocidade constante durante a corrida contra o tráfego de jogo, uma vez que evita a perda de pacotes e os atrasos na cauda causada por picos elevados, sem aumentar o tamanho da janela do remetente. SACK TCP e TCP New Reno, por outro lado, tendem a aumentar continuamente o tamanho da janela de transmissão, o que poderia permitir uma maior perda de pacotes e também para causar atrasos indesejáveis para o tráfego de jogo.

**Palavras chave:** Multiplayer, MCG, realidade virtual.

### Introduction

With the vast field of Internet and wireless communication, multiplayer computer games (mcg) are becoming more popular. The commercially published computer games offer a choice of multiplayer, and at the same time, sites of free online games like Riot games, Electronic Arts, Ultima Online, Blizzard Entertainment's Battle.net or Microsoft MSN playground boast of having hundreds of thousands of users. Similarly, new versions of console games rely heavily on the appeal of online games, and a new branch of the mobile entertainment has

emerged with the intention of developing multiplayer games distributed for wireless applications. In this sense, MCG will continue to provide both technical and practical challenges in the future.

Since the 1980's the United States Department of Defense has developed military applications through a protocol for distributed interactive simulation (DIS), which was published as IEEE standard in 1992. After that, the military research has concentrated on the development of the high-level architecture (HLA), which aims to provide an architecture and services in general for distributed data exchange. While the DIS Protocol is closely linked to the properties of the military units and vehicles, the *raison d'être* of HLA is that it could also be used with not military applications such as computer games.

## Network resources

Distributed simulations face three limitations of resources: bandwidth, latency of the network, and the processing power of hosting to handle the network traffic. These resources are related to the technical attributes of the underlying network and which impose physical constraints that the system cannot overcome and which should be considered in the design.

### Bandwidth

Bandwidth refers to the transmission capacity of a line of communication such as a network. In a few words, the bandwidth is the proportion of the amount of data transmitted or received per unit time. In WANs (wide area networks), the bandwidths range from tens of kbps (bits per second) of dial-up modems up to 1.5 Mbps of T1 and 44,7 Mbps of T3. In LANs (local area networks), bandwidths are much larger ranging from 10 Mbps to 10 Gbps. However, the LAN are limited in size and they support a limited number of users, while WAN networks allow global connections. In addition to the frequency and the messages, bandwidth

requirements depend on the quantity and distribution of users. The first's deployments In LAN environments transmit messages to all participants. Obviously, this leads to problems such as the number of participants grows. Unicast communication between a single sender and a single receiver allows to control and direct traffic. However, since most of the messages are destined to multiple receivers, the unicast wastes bandwidth by sending redundant messages.

During the 1990's appeared the multicasting, which is the communication between a single sender and multiple receivers, has matured as a technique and gradually gained popularity in distributed systems. Multicasting allows the receivers to join groups in which they are interested. The sender only sends one message (as in unicast) to a group, which is received by multiple receivers (as in broadcast) belonging to the group.

### Latency

A network latency indicates the length of time (or delay) that it takes when a message arrives from a designated node to another. In addition, the variance of the latency time (it means phase fluctuation) is another feature affecting interactive applications. The latency cannot be eliminated completely. For example, speed-of-light propagation delays and deceleration of the electrical signal into a single cable only produce a 25 to 30 ms latency across the Atlantic. On the other hand, routing, management of queues and delays in the processing of packages, add tens of milliseconds to the overall latency (for example, in August 2001 an average latency1 round-trip Trans-Atlantic measured was 79.955 ms).

For interactive systems in real time such as MCG, the general rule is that the latency between 0.1 and 1.0 seconds is acceptable. For example, the DIS standard specifies that the network latency should be less than 100 ms00. Latency affects the performance of the user in a non-linearly manner: continuous and fluid Control is possible when the latency does not exceed 200 ms, after which the

interaction becomes more observational and conscious. Accordingly, the threshold of latency when it becomes an inconvenience for the user depends on the type of application. In a Real Time Strategy (RTS), a higher latency (even up to 500 ms) may be acceptable as long as it stays static (it means that the fluctuation is low).

## Classification of games to study

We have two main kinds of multiplayer games that occupy the network at the moment, these are the MMORPG (massively multiplayer online role-playing game) and the FPS (First Person Shooter).

Although the MMORPG are among the most popular games online, both of them have some characteristics in common: the virtual stage is shared by a few tens of players, who use fire guns to kill the enemies or accomplish a mission. The weapon can be improved since the player wins money, depending on their combat abilities.

- The main differences between these two types can be summarized as it follows:
- The session Duration in the MMORPG is longer than in first-person shooter. However, FPS players often play a number of assaults during the same session.
- The number of players that share a virtual stage in a FPS is of a few tens, while thousands of players can simultaneously share an MMORPG virtual world.
- The Real time and interactivity requirements of first person shooters are higher, and the objective of the player is of paramount importance. In contrast, the MMORPG are not based on the good aim, since players first use the mouse to select the target, and then choose the weapon or the curse to use against the enemy.
- First person shooters use the UDP protocol, while most of the MMORPG use TCP.
- 

The last difference is the most important for the present study: the use of UDP in FPSs means no

retransmission when a package is lost. This fact has some implications: for example, a shot can be missed, so players often use bursts of machine guns to kill the enemies. Some games implement packet loss concealment algorithms in order to hide the shortcomings of the network from the players. Latency is of vital importance in these games, in fact, network latency can cause a player to lose a hit that was initially well done.

On the other hand, the MMORPG typically use TCP, which is reliable and prevents the loss of any information related to the actions of the players. When a packet is lost, the Protocol requests a retransmission. But the actions, especially the fights, have some interactivity: the player can select different weapons and curses while the fight is on, and their ability to do this quickly has a significant impact on the final result. So it can be concluded, as reported that these games can be considered as a new kind of service: (soft service) in real time and interactive through TCP.

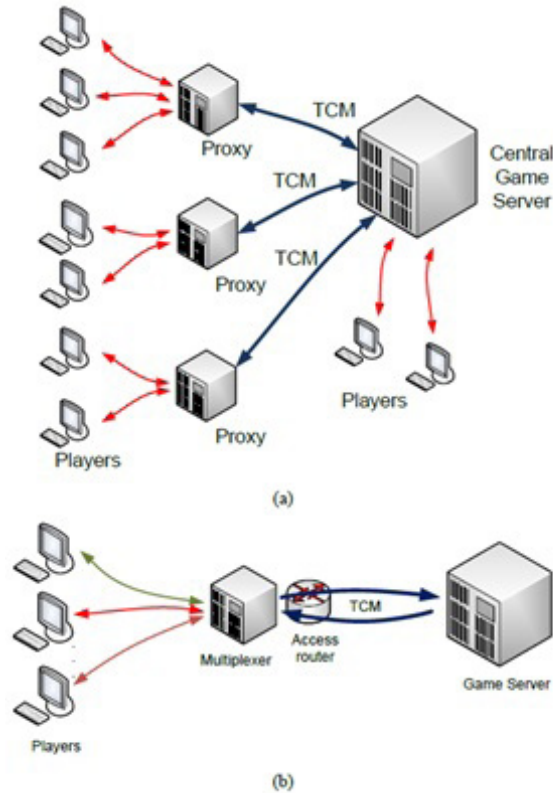
In order to avoid additional delays, the MMORPG usually attaches to the 1 "push" of the TCP header banner, which requires that the package has to be sent as soon as possible. As a result, they tend to generate small TCP packets; where header compression can be applied: we have long-term flows, with many header fields that are the same for each package.

In terms of application scenarios, the technique can be used for the first time between the servers that support the game (Figure 1-a): a proxy may receive the traffic of all users in an area (for example, a city or a district), and forward to the central server, so that the added traffic between the proxy and the server can be compressed and multiplexed, while the addition of small delays.

This can provide a degree of flexibility to the infrastructure support: the bandwidth when it is scarce, the traffic is multiplexed; and when the number of users decreases, the traffic is sent in its native form, thus avoiding additional delays.

Other scenarios, such as a Cyber Cafe (Fig. 1b), where a number of users simultaneously play the same game, may also be suitable for this technique.

However, in this case the number of users can be significantly lower than the one for the stage of server to server.



**Figure 1.** Source: image taken from <http://gamelabstudio.com>.

For MMORPG the traffic occurs through two streams, one carrying the actions of the player (client) to the server, and the other that communicate the current state of the game from the server to the application client. The communication server / client MMORPG does not require a lot of bandwidth; it has been shown that, depending on the situation, traffic demands can increase up to 50 kbps on the downlink and 5 kbps uplink, but these average demands are typically lower. Various models of traffic have been proposed in the literature for the time between packets and (APDU) application protocol data unit sizes.

It was noticed that the game generates small packets, since the majority of the APDU are small and,

in addition, the PUSH bit in the TCP header is set to 1, in order to make the Protocol to send it as soon as possible (instead of waiting for more data to transmit).

## Simulation

One of the most popular MMORPG games is taken in this case, League of legends, a "recent" 2009 game, which has gained great importance in the competitive field.

The generation of traffic simulation in the first place, occurs with a script in NS2, generating APDU with the corresponding statistical distribution for queuing activity in LOL, (eight different values for the client, and a Lognormal distribution for the server), and also with the corresponding times between the arrivals of the APDU (Two Weibull distributions and two deterministic values for the client, and a normal Weibull, and three deterministic values to the server).

### • Unicast routing

ns simple-dyn.tcl

in ns-2/tcl/ex: a simple simulation to illustrate link failure and recovery; no dynamic routing is done to heal the failure.

ns simple-rtg.tcl

in ns-2/tcl/ex: a dynamic routing.

ns simple-eqp.tcl

in ns-2/tcl/ex: equal cost multi-path routing through two equal cost routes.

ns simple-eqp1.tcl

shows equal cost multi-path routing on a different dynamic topology.

### • Multicast routing

ns mcast.tcl

in ns-2/tcl/ex: a multicast routing simulation. Comments in mcast.txt.

ns cmcast.tcl

in ns-2/tcl/ex/newmcast: centralized multicast computation for use in ``session-level'' simulations. Instead of using join messages and implementing multicast routing protocols in the individual routers, the multicast routing tables are implemented in a centralized fashion for the entire topology. PIM sparse mode shared trees and source-specific trees are supported. Comments in cmcast.txt.

```
ns cmcast-spt.tcl
uses source-specific trees.
ns cmcast-100.tcl
```

creates a topology of 100 nodes and 950 edges. 10 members join the multicast group. One sender sends for 90 seconds.

```
ns detailedDM*.tcl
```

in ns-2/tcl/ex/newmcast: Dense Mode protocol that adapts to network changes and works with LAN topologies (LANs created by the multi-link method). Note that this is the recommended version of the dense mode protocol in ns. Comments in detailedDM.txt.

```
ns mcast*.tcl
```

in ns-2/tcl/ex/mcast\*.tcl: multicast routing demos to illustrate the use of Centralized multicast and the dense mode protocols. Comments in mcast.txt

## • Multicast transport

```
ns simple-rtp.tcl
```

in ns-2/tcl/ex/newmcast: RTP. Black for data packets, green for RTCP reports, cream for join messages, purple for prune messages.

ns srm-demo.tcl in ns-2/tcl/ex: an SRM (Scalable Reliable Multicast) demo. Comments in srm-demo.txt.

Five tests demonstrate the behaviour of ns' SRM implementation: Scripts are:

```
ns srm.tcl
```

Different receivers join at different times. Each receiver requests back-data from the start of the source session that it has missed. The demo uses fixed timers.

```
ns srm-star.tcl
```

Code to analyse SRM behaviour in a star topology. Loss of a single packet is created through link failure/recovery. The number of nodes in the star can be changed in the script, but the nam demo is optimised for eight nodes. The demo uses fixed timers.

```
ns srm-chain.tcl
```

Code to analyse SRM behaviour in a chain (or string) topology. Loss of a single packet is created through link failure/recovery. The number of nodes in the chain can be changed in the script, but the nam demo is optimised for eight nodes. The demo uses fixed timers.

```
ns srm-adapt-req.tcl [SrmType]
```

Code to analyse the request parameter behaviour of the SRM Adaptive timers. The topology is a eight-node star. It can be changed in the script. To compare different timer algorithms, the user can specify a different SrmType agent, such as Fixed.

```
ns srm-adapt-rep.tcl [SrmType]
```

Code to analyse the repair parameter behaviour of the SRM Adaptive timers. The topology is a eight-node star. It can be changed in the script. To compare different timer algorithms, the user can specify a different SrmType agent, such as Fixed.

- **Traffic**

ns shuttle-mcast.tcl

in ns-2/tcl/ex: Example multicast traffic on a partial MBone topology

ns web-traffic.tcl

in ns-2/tcl/ex: Example small-scale web traffic

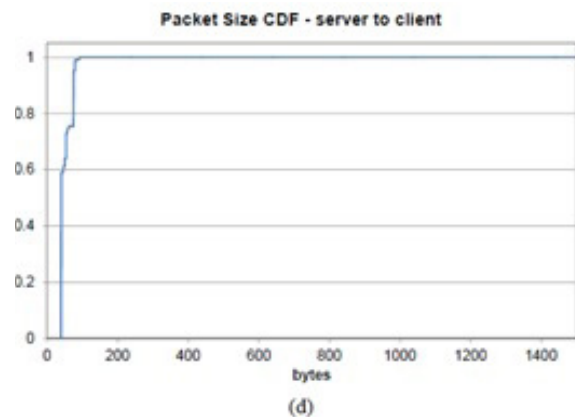
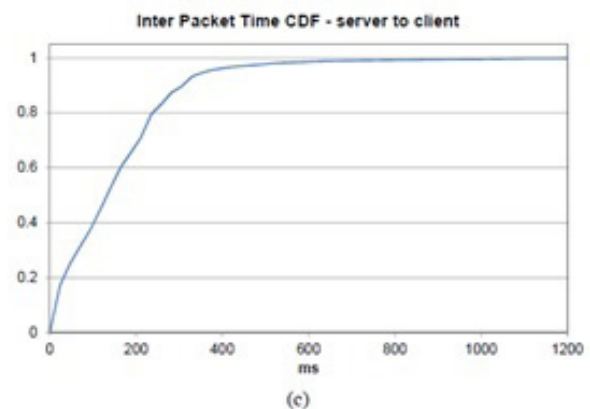
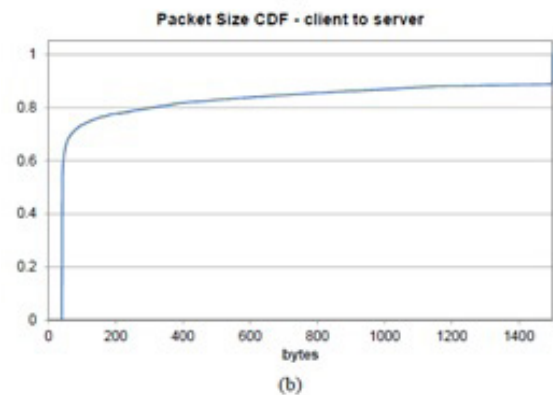
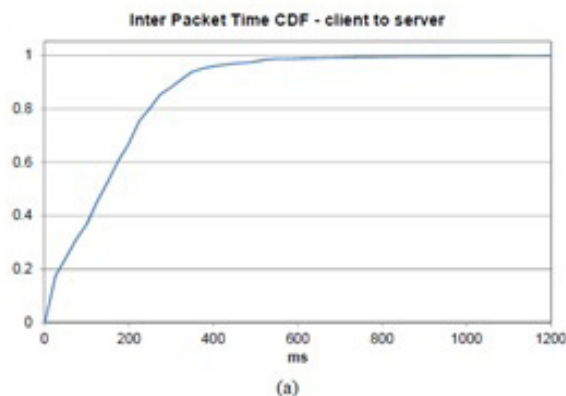
ns large-scale-web-traffic.tcl

in ns-2/tcl/exl: Example large-scale web traffic

The simulated application uses the TCP stack on the client computer (player). For the purposes of simulation, we have selected SACK TCP, since it is a version of TCP that runs on many home computers. The Simulator calculates the inter packet time and APDU size and, if necessary, the APDU is divided into pieces of 1460 bytes, which are shipped in 1500 byte packets. SACK also are sent by the TCP stack, in accordance with the implementation of SACK NS2.

Figure 2 shows the cumulative distribution functions (CDF) for LOL sizes of packages and the times between packages, once traffic has been sent. We can observe that many 40-byte packets are present, corresponding to the ACKs. The time between packets is small, approximately 155 ms on average.

For FTP we use NS2 integrated application, with 1,500 packets of bytes of payload (1,460 bytes) and three variants of TCP, as noted above - SACK TCP, TCP New Reno, and TCP Vegas. The simulation time is set to 1000 seconds.

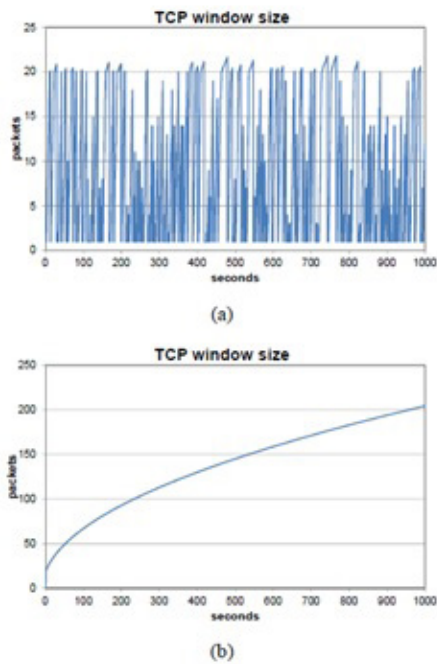


**Figure 2.** Interaction of packages in time and size of the CDF package for the client and the server and the server traffic to client (using TCP SACK).

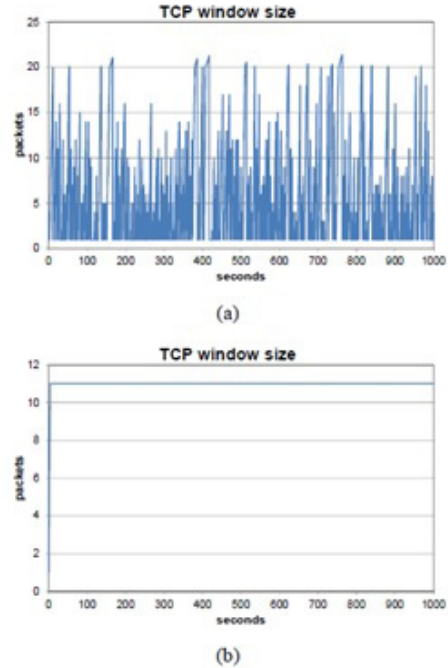
## Results of the simulation

Figures 3 to 5: show the TCP window size for (a) LOL and (b) FTP, for a 200 packet buffer.

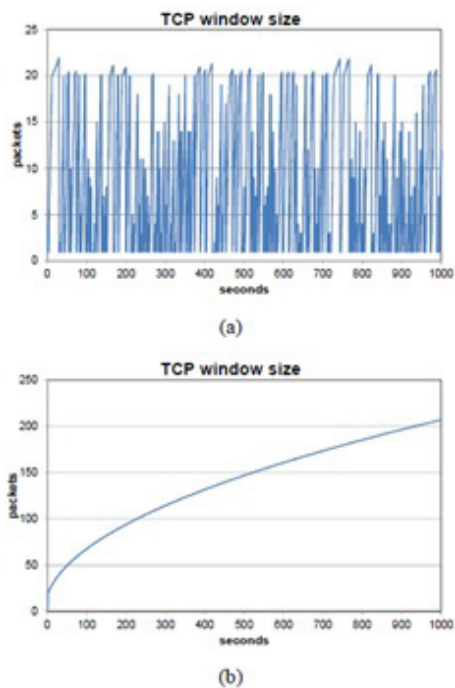




**Figure 3.** TCP window size with a buffer of 200 packages when using TCP SACK for the FTP.  
**Source:** a) LOL; (b) the background traffic.



**Figure 5:** TCP window size with a buffer of 200 packets when using TCP Vegas background FTP.  
**Source:** a) LOL; (b) the background traffic window size.



**Figure 4.** TCP window size with a buffer of 200 packets when using TCP New Reno for the background FTP.  
**Source:** a) LOL; (b) the background traffic.

It can be seen that the size of the window of the TCP New Reno and SACK increase nearly 20% compared to the TCP Vegas, also it shows that even with continuous traffic these two TCPs window is too large which shows a delay of up to 500 ms with little traffic in the background which represents a loss of playability.

Comparing the figures, it is evident that LOL sending window behaves differently from the FTP one. The reason for this is that the application of game is not trying to consume as much available bandwidth as possible (like FTP does). Only has to send a continuous stream of data of small packages, at a speed of less than 10 kbps. Therefore, the size of the window varies, depending on the explosivity of the data generated by the application of the game, its maximum size is about 20 packages.

Seeing this scenario with 200 packages buffer. A 200 package buffer can be considered relatively "large" when it comes to real-time applications, the use of tiny buffers of few tens of kilobytes was



TCP	background traffic window size	background traffic window size	background traffic window size	background traffic window size	background traffic window size	background traffic window size
	100 seconds	300 seconds	500 seconds	700 seconds	800 seconds	1000 seconds
TCP SACK	75 packages	110 packages	149 packages	170 packages	180 packages	200 packages
TCP NEW RENO	75 packages	112 packages	150 packages	170 packages	180 packages	208 packages
TCP VEGAS	11 packages	11 packages	11 packages	11 packages	11 packages	11 packages

recommended to maintain the delay at its minimum. Figures. 3 and 4 show that the FTP window grows continuously when using TCP SACK and TCP New Reno. While a larger buffer in the uplink has a positive effect in the prevention of the occurrence of packet loss on the downside also adds delays which are unacceptable from the perspective of the players as shown in [2]. When using TCP Vegas for background flow (Fig. 5), however, the window size remains stable, this "shy" behavior is caused by the increase in  $RTT^*$ , so it keeps its shipping time at the same rate, even when there is no packet loss.

## Discussion of results

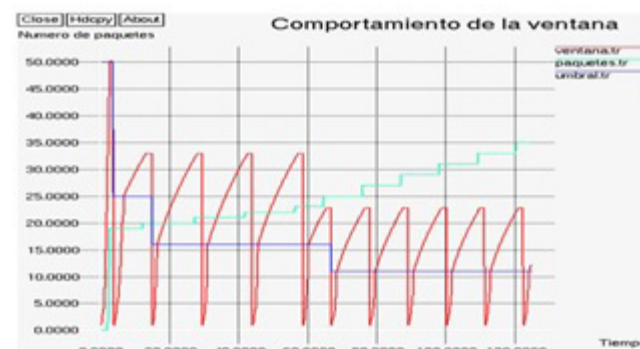
To compare the results obtained with other works first, we took the results obtained by:

Nancy Yaneth Gelvez Garcia master in information science and communications, Andrés Rogelio Córdoba Ramirez and Jairo Andrés Supelano Rátiva Students of electronic engineering all from the Distrital University Francisco José de Caldas where they presented a paper entitled: "COMPARISON BETWEEN THE TYPES OF TCP RENO, SACK AND VEGAS FROM THE POINT OF VIEW OF THE CONTROL OF FLOW" where it shows in depth these three types of tcp's and clearly state which

of them is better in flow control, here they show the loss of packets in contrast to the window size.



**Figure 6.** Window size, lost packets and threshold with TCP RENO with FTP traffic, times, packages.



**Figure 7.** Window size, lost packets and threshold with TCP SACK with FTP traffic, times, packages.



**Figure 8.** TCP Window size, lost packets and threshold with TCP VEGAS with FTP traffic, times, packages.

Results are similar to those obtained in the tests showing that the loss of packets with TCP vegas is much lower compared to TCP New Reno, and TCP SACK.

## Conclusions

The results show that TCP Vegas is able to maintain a constant speed while racing game traffic, since it avoids the loss of packets, avoiding the increase in size of the shipping window. Against that, TCP SACK and TCP New Reno tend to continue increasing the size of the window, and so the addition of the delays for the game traffic also increases. Finally, smaller buffers have proven to be better for MMORPGs based on TCP, since larger buffers cause bigger delays.

## References

- B. Thomas. (1998). *Comparison of TCP Reno and TCP Vegas via fluid approximations*. Institut National de Recherche En Informatique Et En Automatique.
- Gélvez, N., Córdoba, A. & Supelano, J. (2012). Comparación entre los tipos de tcp reno, sack y vegas desde el punto de vista del control de flujo. *Redes de Ingeniería*, Vol. 2(2).
- Huang, G., Ye, M. & Cheng, L. (2004). *Modeling System Performance in MMORPG*. In: *Globe-com 2004 Workshop*, IEEE.
- Miller, J.L., Crowcroft, J. (2010). *The Near-Term Feasibility of P2P MMOGs*. In: *Proc. International Workshop on Network and Systems Support for Games (NetGames)*.
- Ries, M., Svoboda, P. & Rupp, M. (2008). *Empirical study of subjective quality for Massive Multi-player Games*.
- Vishwanath, A., Sivaraman, V. & Rouskas, G. N. (2009). *Considerations for Sizing Buffers in Optical Packet Switched Networks*. In: *IEEE INFOCOM*, Brazil.
- Wu, Y., Huang, H. & Zhang, D. (2006). *Traffic Modeling for Massive Multiplayer On-line Role Playing Game (MMORPG) in GPRS Access Network*.

